

BOG 12:

# Programming Environments: Debugging, Autotuning, and Specialization

ASCR Workshop on Extreme Heterogeneity in HPC  
23-25 Jan 2018

# BOG 12 Contributors

Moderator(s): Mary Hall, John Mellor-Crummey

Rich Lethin, Abid Malik, Sam Williams, George Biros, Stanimire Tomov, Brian Van Straalen, Khaled Ibrahim, Michael Garland, Pat McCormick, David Bernholdt, David Donofrio, EJ Park, Esam El-Araby, Jeremy, Lingda Li, Meifeng Lin, Michelle Strout, Bob Lucas, Rich Lethin, Sameer Shende, Scott Baden, Seyong Lee, Sonia Sachs, Sherry Li

# BOG 12 Current Capabilities

Capability 1: Hand-crafted domain-specific environments that leverage application knowledge to generate high performance code for CPUs and accelerators

Capability 2: Autotuning technology that achieves high performance through offline exploration of a large but narrowly defined search space

Capability 3: Instrumentation-based tools to detect memory errors and data races

Correctness guarantees, overhead, and scalability vary based on the tool

Capability 4: Debugging, including reversible debugging based on snapshots and event logs

Capability 5: Hardware support for performance measurement

Event counts, threshold and instruction-based sampling, tracing

# BOG 12 Possible Research Directions Summary

PRD 12.1 - Formal methods and tools for construction of correct parallel programs

PRD 12.2 - Compiler technology, DSLs, autotuning and software infrastructure for dynamic optimization across a range of architectures (single-source performance portability)

PRD 12.3 - Hardware and software mechanisms for monitoring correctness in EH components

PRD 12.4 - Hardware and software mechanisms for online performance introspection and analysis

# BOG 12 Capability Targets for Extreme Heterogeneity

BOG 12 brainstorming and discussion of capabilities that will be needed in the 2025-2035 timeframe to make increasingly heterogeneous hardware technologies useful and productive for science applications.

Correct and efficient code for EH architectures

Automation of translation from single-source implementation to maximize human productivity

- Starting from expressive high-level specifications

# BOG 12 Capability Targets for Extreme Heterogeneity

- Software systems that aid in generating or proving code correctness for platforms that support massive heterogeneous parallelism
- Hardware and software mechanisms for correctness introspection within and across components in workflows on EH platforms
- Hardware and software mechanisms for identifying and quantifying opportunities for performance improvement in workflows on EH platforms
- Dynamic code generation of high-performance code for EH platforms
  - for individual hardware components in isolation
  - for multiple coupled components

# BOG 12 Targets for 2030

Target 1:

Target 2:

# BOG 12 Current Status

BOG 12 survey describing the current status of science, technology, or practice related to this theme (starting with material directly from the FSD).



# Domain-Specific Specialization

Two approaches:

- Domain-specific environments by application community are tailored to application teams (e.g., TCE, Chombo, MADNESS)
- More general domain-specific tools developed by research community (e.g., Polymage, Pochoir) or commercial enterprises (e.g., Halide, TensorFlow)

Limitations

- Composability across domains
- Generality
- Maintainability

Broader adoption would generate a community, ensure longevity

# Autotuning

Motivated by difficulty in modeling complex architectures -- use empirical data

Most common use:

- Tuned library or tuned application kernel (source, binary) code integrated into application
- Offline autotuning search or training for machine learning model explores search space of alternative implementations

Limitations

- Changes to execution context (new hardware, new compiler/runtime/OS, different load) invalidate offline experiments
- Often significant manual effort to set up autotuning experiments and significant compute resources used in search
- Instability of measurements at scale may misguide search

Integrating into the build/link process, and transparency from the user would increase adoption

# Correctness

Synthesis of provably correct code: DeepSpec

Model checking: exhaustive examination of finite state systems

Dynamic binary instrumentation tools (e.g., Valgrind)

- memory state analysis (free, allocated, initialized)

Data race detection

- Instrumentation approaches

  - generation of instrumented code (e.g. LLVM's TSAN)

  - dynamic binary instrumentation (Helgrind, Cilkscreen, ARCHER)

- Dynamic approaches: access histories in shadow memory, reachability analysis, labeling strategies

Debuggers, including reversible execution

Hardware support: transactional memory, watchpoints

# Performance Analysis

## Measurement methodologies

- event counters

- instruction-based sampling, marked instructions

- Instrumentation

- top-down methodologies

- call stack collection

## Profiles and traces

Attribution to source-level view with compiler & runtime support (e.g. OMPT)

Visualization of space-time diagrams: message traces, sample traces

## Automated analysis

- clustering

- pattern identification and root cause analysis using traces

# BOG 12 Challenge Assessment

Discussion to identify research required to get from where the capabilities are now to where to where they need to be by 2030.

- R&D of software frameworks that support construction of domain-specific environments
  - Rule-based methods for tailoring workflows to one or more components in EH platforms
- Investigation of autotuning based on dynamic feedback-directed optimization
  - Autotuning as part of the application build/link process to adapt to execution context
- R&D of methods for pinpointing causes of performance losses, assessing utilization, and identifying rate limiting factors

# BOG 12: List of Key Research Challenges

Challenge 12.1 Develop programming abstractions and compiler technology that insulates application developers from the complexity of the memory subsystem and compute engines on EH platforms yet helps them exploit EH platforms effectively

Challenge 12.2 Develop hardware mechanisms that support correctness and performance introspection within and across components of EH platforms

# PRD 12.1 : Short title of possible research direction

- One paragraph description (3 sentence/bullet)
- Research challenges
  - Metrics for progress
- Potential research approaches and research directions
- How and when will success impact technology?